



Nama Formulir:

**Lembar
Pengesahan
Karya Ilmiah**

No. FR-003/PR-003/KB-02-01/MMP/UPM/2020

Issue/Revisi 1

Tgl Berlaku 15 Juli 2020

Halaman 1

Yang bertanda tangan di bawah ini:

1. Nama Lengkap Wahyuningdiah Trisari Harsanti Putri, M.T.I
Jabatan Ketua Program Studi
Program Studi Informatika
NIP 214040276

Telah melakukan penelaahan karya ilmiah berjudul:

Membangun Sistem Peningkatan Basis Data Menggunakan Metode Sharding

Dan menyatakan bahwa karya ilmiah tersebut sudah memenuhi kaidah penulisan ilmiah dan oleh karenanya layak diajukan untuk keperluan unggah karya ilmiah di Universitas Paramadina, atas nama:

Nama Lengkap Muhamad Albi Pradana Rizal Fadli
Jenjang S1
Program Studi Informatika
NIM 120103013

Demikian hasil penelaahan atas karya ilmiah ini dibuat dengan sesungguhnya untuk dapat dipergunakan bilamana diperlukan.

Jakarta, 16 Februari 2023

Penelaah,

NIP: 214040276

MEMBANGUN SISTEM PENINGKATAN BASIS DATA MENGGUNAKAN METODE SHARDING

Muhamad Albi Pradana Rizal Fadli
Universitas Paramadina
muhamad.fadli1@students.paramadina.ac.id

Abstract

This research is motivated by the rapid development into the digital era. As this simple factor that will drive a considerable for the traffic increased in a system, then this is drives for large storage needed and required proper capable database system management. An alternative way for this problem is horizontal architecture. horizontal architecture is a method that separated data into different machines. As the requirement of database type to supporting this method is mongoDB within implement sharding and replication method. The result of this research is the sharding method using a NoSQL database with the mongoDB type is an efficient solution for implement horizontal database system proses. The amount of data storage will be aligned with the number of clusters owned by a system.

Keywords: *Sharding, Replication, mongoDB, NoSQL, Horizontal*

Abstrak

Penelitian ini dilatarbelakangi oleh perkembangan ke era digital yang sangat cepat. Dari faktor yang cukup sederhana tersebut mendorong perubahan yang cukup besar terhadap kenaikan lalu lintas data dalam sebuah sistem, kemudian ini juga yang mendorong kebutuhan media penyimpanan yang cukup besar dan juga serta memerlukan pengelolaan sistem Basis data yang sangat mumpuni. Alternatif dari permasalahan ini adalah membutuhkan arsitektur yang metode pertumbuhannya berfokus pada peningkatan yang menyebar atau sering disebut partisi horizontal. Sedangkan partisi secara horizontal adalah metode yang memecah data dengan batasan baris ke dalam mesin-mesin yang berbeda. Kemudian dengan kebutuhan Jenis Basis data yang mendukung metode ini adalah mongoDB dengan metode replikasi dan sharding. Hasil dari pengujian ini ialah metode sharding dengan menggunakan basis data NoSQL dengan jenis mongoDB merupakan solusi yang efisien dalam proses peningkatan sistem basis data horizontal. Besaran penyimpanan data akan selaras dengan jumlah kluster yang dimiliki oleh sebuah sistem.

Kata kunci: *Sharding, Replication, mongoDB, NoSQL, Horizontal*

1. Pendahuluan

Pada era digital ini Internet berkembang pesat dan akan menjadi salah satu kebutuhan utama untuk beberapa era ke depan. Oleh sebab itu perkembangan internet yang cukup pesat ini memicu perusahaan yang bergerak pada bidang teknologi dan informasi untuk selalu melakukan pemutakhiran sistem yang di milikinya. Dari

faktor yang cukup sederhana tersebut mendorong perubahan yang cukup besar terhadap kenaikan lalu lintas data dalam sebuah sistem, dan ini juga yang mendorong kebutuhan media penyimpanan yang cukup besar dan juga serta memerlukan pengelolaan sistem Basis data yang sangat mumpuni, jumlah data yang cukup besar ini akan berdampak pada data yang disimpan maupun

sistem pengelolaannya yang harus sejalan dengan pertumbuhan data [1]. Permasalahan yang akan diangkat pada karya akhir ini ada peningkatan sistem Basis data tidak selalu mudah dilakukan karena mahal biaya untuk memperoleh sumber daya tersebut. Alternatif dari permasalahan ini adalah scaling-out-database atau lebih dikenal sebagai arsitektur yang metode pertumbuhannya berfokus pada peningkatan yang menyebar dengan metode ini dapat melakukan pengelompokan beberapa mesin kecil dalam sebuah kluster [1]. kemudian pada karya akhir ini akan didasari ada dua metode yang digunakan untuk mendukung kapabilitas ini yaitu replikasi serta sharding. sharding dan replikasi adalah sebuah metode yang saling berkaitan pada sebuah sistem Basis data, sistem ini ialah pola arsitektur Basis data yang berkaitan dengan teknik memisahkan satu baris tabel menjadi beberapa tabel berbeda, yang dikenal sebagai partisi data atau pemecahan data merupakan sebuah arsitektur yang dapat digunakan memisahkan data menjadi lebih kecil [2]. Pengembangan Basis data yang berorientasi pada metode horizontal ini merupakan salah satu model Basis data yang tidak ketergantungan di dalamnya dapat disebut dengan non-relasional Basis data. Pada model ini data disimpan dengan menggunakan sistem replikasi yang menyebarkan data ke beberapa kluster - kluster kecil. Kemudian dengan kebutuhan Jenis Basis data yang mendukung metode ini salah satunya adalah mongoDB sehingga penerapan metode replikasi dapat di mungkin untuk digunakan pada Karya Akhir ini. 8 Pada Karya Akhir ini, penulis akan melakukan penelitian untuk membangun sistem Basis data yang menggunakan infrastruktur horizontal pada sebuah kluster dengan mengintegrasikan data dengan beberapa kluster lain pada sebuah lokal server, serta setelah melakukan perubahan sistem Basis data menggunakan metode ini apakah kemampuan sistem untuk selalu bertumbuh akan lebih baik.

2. Landasan Teori

2.1. Definisi Basis data

Basis data pada umumnya digunakan sebagai media penyimpanan kumpulan - kumpulan dari beberapa data yang terhubung dan terstruktur sebagaimana mestinya dengan tujuannya adalah untuk mempermudah pengelolaan data yang jumlahnya ribuan hingga jutaan data yang data tersebut terus bergerak untuk menyuplai data pengguna [3]. Di dalam basis data ini tidak hanya terdapat kumpulan data, tetapi juga bisa terdapat metadata. Pada zaman ini rata - rata orang sudah mulai bergantung pada aplikasi digital. Keefisienan aplikasi tergantung pada sistem basis datanya dan menjadi salah faktor penting pada suatu aplikasi untuk melakukan pengolahan suatu data dalam jumlah besar. Kemampuan dalam melakukan pengolahan data yang menjadi kunci utama dipilihnya Basis data menjadi komponen penting tersebut. Di dalam Basis data penyimpanan dilakukan secara terintegrasi, berbeda dengan sistem file.

2.2. NoSQL Basis data

Berdasarkan dengan penamaan NoSQL, ini dapat merepresentasikan jenis Basis data ini sebagai Not Only SQL [4]. Secara harfiah NoSQL adalah sebuah basis data yang tidak terdapat perintah-perintah SQL yang konsep penyimpanan data pada basis data ini yang memiliki jenis terstruktur, semi terstruktur, atau bahkan tidak terstruktur [4]. Jenis basis data ini sangat sesuai dengan pengembangan aplikasi terkini keran terbukti mampu menangani data pada jumlah besar dan berjalan pada kluster. Pada penelitian Karya Akhir ini akan menggunakan basis data jenis ini sebagai sarana baru untuk menggunakan basis data dengan tujuan yang lebih efisien untuk mengetahui performa pengolahan data pada penelitian ini.

2.3. Document oriented Basis data

Menurut Oliver Schmitt tujuan dasar dari basis data yang berbasis dokumen adalah bahwa unit terkecil untuk penyimpanan Adalah dokumen [5]. Struktur data dari dokumen dapat terdiri atas

pasangan Key-value. Beberapa jenis Basis data yang mendukung document-oriented Basis data adalah couchdb , mongodb, orientdb, ravendb dan terratsore.

2.4. MongoDB Basis data

MongoDB adalah salah satu jenis Basis data manajemen yang didesain untuk berjalan di internet dan aplikasi berbasis web, ini adalah juga merupakan jenis Basis data berbasis document atau dapat juga disebut NoSQL Basis data [6]. Data model yang digunakan pada jenis Basis data ini memiliki kemampuan baca dan tulis tinggi untuk peningkatan yang lebih dengan failover otomatis. MongoDB dengan model data document lebih mudah untuk di bangun, karena memiliki dukungan yang untuk jenis data yang tidak terstruktur dan tidak memerlukan migrasi yang mahal dan memakan waktu pada saat terdapat pada sisi aplikasi.

MongoDB dengan model data dokumen di bangun untuk menggunakan format mirip seperti JSON, yang sering disebut juga BSON. BSON sangat cocok untuk arsitektur pemrograman yang berorientasi objek modern, ringan, dan cepat [6]. MongoDB menggunakan BSON sebagai format untuk media bertukar data melalui. Basis data MongoDB memahami internal BSON. Ini berarti bahwa MongoDB dapat menjangkau di dalam objek BSON, bahkan objek yang lebih kecil dalam sebuah dokumen, dan menggunakan delimiter notasi titik. Ini memungkinkan MongoDB membangun indeks dan mencocokkan objek dengan bahasa query pada kunci BSON tingkat atas dan bertingkat. MongoDB juga mendukung query dengan jumlah yang besar serta indeks. Ini membedakannya dari Basis data dokumen lain di mana lapisan server terpisah digunakan untuk menangani query kompleks [6]. Fitur lainnya termasuk sharding otomatis, replikasi, dan penyimpanan yang mudah.

2.5. Document Oriented Storage

MongoDB melakukan penyimpanan dalam bentuk JSON format document pada semua atribut [6], yang ini merupakan kelebihan basis

data jenis karena menggunakan format JSON maka tidak memerlukan relasi antar table.

2.6. Replication & High Availability

MongoDB mampu menggunakan metode replikasi dengan menggunakan beberapa kumpulan dari replikasi data [6]. Kumpulan replika adalah gabungan beberapa mongod yang memberikan penamaan kumpulan data yang sama. Dalam sebuah replika data terdiri dari satu node utama yang digunakan untuk menerima semua operasi penulisan data untuk melakukan penyimpanan. Kemudian juga terdiri node kedua, yang fungsi melanjutkan operasi dari node utama sehingga mereka memiliki kumpulan data yang sama.

2.7. Auto-sharding

Sharding adalah proses melakukan penyimpanan data record yang berjalan melalui beberapa mesin dan ini adalah pendekatan yang digunakan oleh MongoDB untuk menyatukan kebutuhan data yang berkembang, sesuai dengan ukuran dari sebuah data yang bertambah, sebuah mesin mungkin tidak akan cukup untuk menyimpan data yang diberikan untuk membaca dan menuliskan data [6]. Sharding menemukan jalan untuk memecahkan masalah dengan menggunakan peningkatan horizontal, dengan metode ini kita dapat menambahkan mesin untuk mendukung pertumbuhan data serta kebutuhan untuk proses pembacaan dan penulisan.

2.8. Perancangan Basis data

Dalam suatu Basis data untuk melakukan perancangan diperlukan beberapa fase penting yang harus dilakukan. Adapun tahapan dalam proses merancang basis data dibagi menjadi tiga bagian yang pertama adalah proses perancangan basis data konseptual, yang kedua adalah proses perancangan secara logika dan yang ketiga adalah proses perancangan secara fisik. Perancangan basis data yang dilakukan secara konseptual adalah kegiatan membentuk permodalan data yang digunakan pada suatu instansi, dan juga tidak terikat pada semua ketentuan fisik. Pada proses perancangan basis data secara logika

kegiatan dilakukan pembuatan permodalan data yang dipakai dalam organisasi yang bersumber berdasarkan bentuk data yang rinci, namun bersifat bebas tidak terikat dengan DBMS (Basis data Management systems) dan terbebas juga dari ketentuan fisik apa pun. Hasil dari tahap ini berbentuk kamus data yang di dalamnya terdapat atribut beserta key (primary key, alternate key, foreign key) serta entity relational diagram (ERD). Perancangan basis data fisik merupakan suatu kegiatan dengan tujuan menjabarkan penerapan basis data pada penyimpanan kedua, penggambaran relasi dasar antar data, kumpulan dari beberapa file, dan indeks yang menjadi acuan tercapainya jalur yang efisien untuk data, dan semua masalah keutuhan yang berkaitan dan termasuk pada tahap-tahap keamanan (security). Melakukan proses perancangan Basis data bertujuan untuk melengkapi informasi yang menjadi kebutuhan-kebutuhan pengguna baik yang mempunyai sifat terbatas beserta beberapa aplikasinya, memberikan kemudahan penjelasan rangkaian dari informasi, membantu meringankan kebutuhan pembentukan proses begitu juga beberapa objek penampilan (response time, processing time, dan storage space)

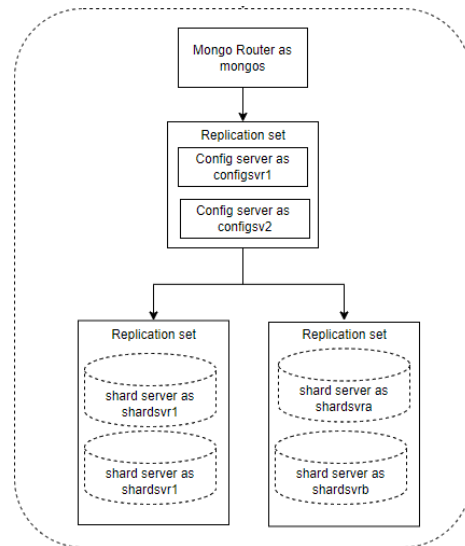
3. Metode Penelitian

metodologi yang akan digunakan pada penelitian ini yaitu membangun sistem basis data dengan menggunakan metode sharding berbasis *NoSQL*, dengan tujuan menerapkan fungsi yang terdapat pada metode tersebut yang telah ada sebelumnya. Sedangkan dalam fase pengumpulan data, penelitian ini akan menerapkan pendekatan kuantitatif berdasarkan hasil keluaran dari sistem yang dibangun.

3.1. Diagram Alir Sharding Basis data

Diagram alir sharding basis data adalah tahapan berikutnya setelah melakukan analisis pada kebutuhan dan batasan sistem. Pada tahapan ini peneliti akan menjelaskan sistem yang akan dibangun berupa alur sistem yang akan berjalan menggunakan diagram alir untuk memaparkan

sistem *sharding basis data*. Tujuan dari tahap ini yaitu menghasilkan rancangan sistem yang dapat digunakan pada bab implementasi sistem. Sistem rancang aplikasi sharding basis data yang dikembangkan akan tampak pada gambar 3.1 berikut.



Gambar 3.1 Diagram Alir sharding basis data

3.1.1. Api Input

Server ini berfungsi sebagai media klien melakukan proses request insert data pada sebuah lokal server. Pada sistem yang di bangun pada penelitian ini akan menggunakan data berupa buku pada sebuah perpustakaan.

3.1.2. Mongos Router

Server ini berfungsi sebagai komponen penting pada sebuah basis data yang berbasis *NoSQL* khususnya untuk jenis *mongoDB* yang tujuan utama dari server ini adalah menampung request klien dan sebagai media untuk mengolah data request tersebut ke arah shard server, di sini semua config server untuk mendukung metode sharding [7]. Pada server ini juga terdapat console untuk melakukan pengecakan data request akan disimpan dan server juga bersifat single server yang tidak miliki replikasi server seperti pada gambar 3.5 di atas. Kemudian fungsi utama selanjutnya adalah sebagai *controller* data ke arah shard server apakah data dengan nilai yang telah

di tentukan akan masuk ke shard 1 ataupun shard 2 sesuai dengan jenis sharding yang diambil.

3.1.3. Config server

Server ini di fungsikan sebagai data config mongoDB karena keseluruhan config yang terdapat pada Mongos router dan shard server akan di simpan pada server ini, tujuan utama pada di server ini adalah untuk menghubungkan Mongos router dan shard server sesuai dengan config yang telah di sesuaikan [7]. Di dalam server di perankan server sebagai configsvr yang dan juga terdapat replikasi oleh dua buah server.

3.1.4. Shard server

Server shard merupakan jenis server yang digunakan sebagai media untuk penyimpanan data [7]. Pada server ini jumlah server kluster dapat di ubah sesuai dengan kebutuhan sistem artinya untuk keperluan skala pembesaran, server shard dapat tambah sesuai dengan jumlah data yang meningkat pada sistem. Pada aturan baku sistem mongoDB setiap kluster seminimalnya memiliki tiga buah server replikasi pada setiap klusternya, jadi ketika mengalami masalah antar koneksi maupun Hardware di salah satu server shard yang mana masih terdapat dua buah server standby untuk menggantikan server tersebut dan aturan bawaan mongoDB server kluster tidak akan bisa berjalan jika hanya satu yang tersisa, namun untuk mempermudah pengujian pada karya akhir ini setiap kluster hanya memiliki dua buah server karena semakin banyak server maka perlu Resource yang semakin besar pula.

3.2. Desain Basis data

Berikut di bawah ini adalah pemodelan dari basis data MongoDB yang akan menerapkan replikasi dan sharding pada sistem basis data. Pada tahapan ini akan memberikan penjelasan nama basis data dan collection. Basis data mongoDB ini akan mengimplementasi sharding dengan metode hashed yang memecah data dengan menggunakan nilai dari hashed tersebut. pada gambar desain basis data perpustakaan sebagai basis data dan buku sebagai collection yang di dalamnya terdapat kolom *_id*, *compelated*,

createdAt, *bukuId*, dan *description*. kemudian hashed dapat disebut sebagai shard key atau kunci utama dari sharding basis data yang fungsinya sebagai map lokasi data akan di sebar ke shard mana [7]. Pada basis data ini bukuId di fungsikan sebagai shard key untuk keseluruhan data pada basis data perpustakaan. Shard key ini juga dibutuhkan ketika proses query yaitu *select*, *insert* dan *update*.

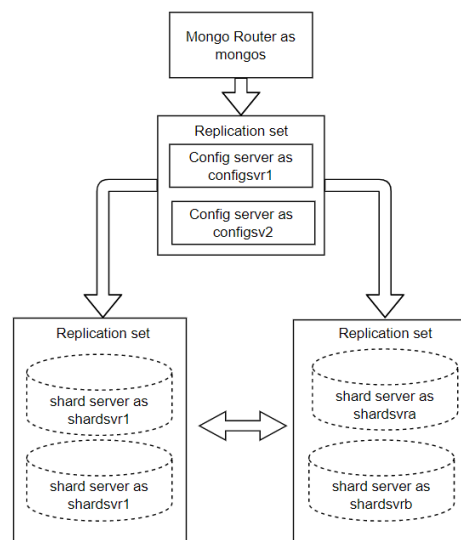
4. Implementasi dan pengujian

4.1. Implementasi Sistem

Sebelum melakukan implementasi perangkat lunak, maka perlu dibuat terlebih dahulu server virtual pada aplikasi VMware dengan sistem operasi yang digunakan adalah CentOS dan basis data menggunakan mongoDB.

4.1.1. Implementasi & konfigurasi MongoDB

Pada proses implementasi dan konfigurasi akan menjelaskan bagaimana sistem terintegrasi dengan beberapa kluster yang akan digunakan dalam proses pengujian sharding dan replikasi basis data. MongoDB telah mendesain sistemnya untuk setiap kluster harus mempunyai minimal 2 server sebagai replikasi yang akan di integrasikan dengan node yang sama. Agar membantu untuk menjelaskan bagaimana struktur utama sistem akan tampak pada gambar 4.1 di bawah ini.



Gambar 4.1 konfigurasi mongoDB

4.1.2. Konfigurasi server *Shard*

Server shard ini akan digunakan sebagai media penyimpanan data. Pada proses konfigurasi ini akan dijelaskan bagaimana sistem akan diintegrasikan dengan satu shard dengan shard yang lain dengan metode replikasi set basis data atau lebih umum dikenal sebagai kloning basis data sebagai server cadangan ketika terdapat kerusakan di salah satu server shard, kemudian replikasi set ini akan diintegrasikan antar replika set menggunakan metode sharding basis data yang gunanya untuk mengatur lalu lintas data dengan metode ini pula horizontal skala pada basis data dapat dilakukan.

4.1.3. Konfigurasi Server Config

Config server adalah sistem yang digunakan mengatur lalu lintas data antar sharding dan yang akan menghubungkan shard dengan Mongo router. Pada karya akhir ini pula config server akan memiliki 2 buah server yang perlu diintegrasikan terlebih dahulu dengan metode replikasi.

4.1.4. Konfigurasi Server Mongo router

Mongos merupakan server utama dari sistem mongoDB yang gunanya untuk menghubungkan klien server dan basis data server. Pada server terdapat informasi antar server replikasi, sharding, *db name*, *collection name* yang saling terintegrasi. sistem ini akan tampak pada Mongo client atau *admin console*.

4.2. Pengujian Sistem

Pengujian pada sistem basis data mongoDB akan berfokus pada sharding dan replikasi data yang merupakan pokok permasalahan karya akhir ini. Sistem sharding akan melakukan proses pemecahan data ketika klien server melakukan proses input data terhadap basis data sedangkan replikasi akan mengintegrasikan server antar node dalam 1 kluster

4.2.1. Pengujian server sebagai Replikasi

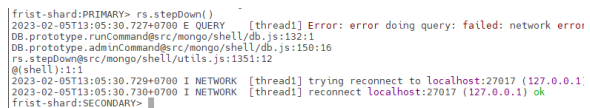
Setelah melakukan proses konfigurasi untuk kebutuhan replikasi data, maka akan dilanjutkan dengan proses pengujian antar replikasi data dengan Multi-Basis data skema yang

menghubungkan replikasi pada kluster first-shard dan second-shard. Pengujian utama replikasi adalah ketika ada masalah putus koneksi antar server replikasi di server primary kemudian yang salah satu server cadangannya akan di alihkan sebagai primary server. Untuk mendukung pengujian ini dibutuhkan sistem monitor ketika peralihan dari secondary menjadi primary basis data pada setiap kluster, salah satu media memonitor yang dapat digunakan untuk pengujian ini adalah menggunakan program dari nodeJs. Tampilan utama dari monitor akan menampilkan informasi mongoDB dan konfigurasi akan tampak pada gambar 4.2 di bawah ini.



Gambar 4.2 Logging sistem sharding basis data

Pada pengujian pertama replikasi server shardsvr2 dan shardvr1 difungsikan sebagai primary basis data di mana semua data input/output dari klien akan diolah di dalam server tersebut yang berarti shardsvr2 dan shardsvr3 adalah sebagai secondary basis data atau cadangan basis data. Pada setiap server pada kluster yang sama akan memiliki data yang sama namun di kluster lain akan memiliki data yang berbeda antar kluster second-shard dan first-shard. Pengujian selanjutnya adalah ketika salah satu dari primary server akan dimatikan melalui *mongo console* menggunakan perintah *rs.stepDown()* akan tampak pada gambar 4.3 di bawah ini.



Gambar 4.3 Proses stepdown 1 server

Kemudian shardsvr2 akan otomatis menjadi primary basis data pengganti shardsvr1 dan shardsvr2 akan menjadi secondary dapat di lihat pada gambar 4.3 di bawah ini untuk proses perubahan primary ke secondary dan sebaliknya

```

Time : 2023-02-05T06:06:52.537Z
Connection : mongodb://127.0.0.1:27017/perpustakaan
Configuration : sharded

Shard: second-shard
PRIMARY : shardsvr1:27017
SECONDARY : shardsvr2:27017

Shard: frist-shard
SECONDARY : shardsvr1:27017
PRIMARY : shardsvr2:27017

```

Gambar 4.4 Monitor sistem replikasi

Pada pengujian replikasi basis data di atas dapat di lihat perubahan secondary dan primary telah memenuhi syarat sebagai replikasi yang terintegrasi antar 1 kluster yang sama. Metode ini dapat membantu sebuah sistem yang sedang berjalan pada lalu lintas data tinggi maupun sesi aktif tanpa menghentikan layanan tersebut dan kinerja dari sistem terlebih dahulu ketika sedang mengalami gangguan koneksi atau problem Hardware pada salah satu servernya.

4.2.2. Pengujian server sebagai Sharding

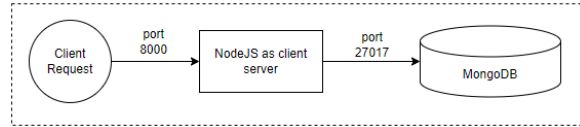
Setelah melakukan pengujian replikasi pada tahapan sebelumnya berhasil menghubungkan antar server di satu kluster yang sama, maka pada tahapan ini akan membahas bagaimana proses pengujian sharding pada basis data mongoDB yang telah di konfigurasi pada tahapan konfigurasi sebelumnya. Pada pengujian ini akan berupa hasil setelah mengonfigurasi sharding apakah data akan sesuai dengan hasil yang dibutuhkan. Pengujian utama pada tahapan ini adalah memantau data ketika proses input data dari klien server untuk mendukung pengujian ini maka perlu ada sistem pihak ketiga. Aplikasi yang membantu proses sharding telah sesuai yaitu nodeJs sebagai pendukung REST-API ke arah basis data server. Api yang di sediakan pada pengujian ini adalah api insert dan select kemudian request akan di kirim langsung ke mongoDB melalui mongo server klien. Pengujian pertama klien akan melakukan insert data melalui Api insert dari server klien menggunakan Port 8000 pada lokal server akan tampak pada gambar 4.5 di bawah ini.

```

frskt@mongos -> curl -X POST http://127.0.0.1:8000/todos -H "Content-type: application/json" -d '{"bookId": "algoritma_dasar", "description": "25", "completed": true}'
{"_id": "63e6a8ace5889660172469", "bookId": "algoritma_dasar", "description": "25", "completed": true, "createdAt": "2023-02-06T10:46:34.991Z"}
frskt@mongos ->
frskt@mongos -> curl -X GET http://127.0.0.1:27017/perpustakaan/todos -H "Content-type: application/json"
{"_id": "63e6a8ace5889660172469", "bookId": "algoritma_dasar", "description": "25", "completed": true, "createdAt": "2023-02-06T10:46:34.991Z"}
frskt@mongos ->

```

Gambar 4.5 Api request klien



Gambar 4.6. Arsitektur Port Sharding

Pada pengujian di atas pada gambar 4.12 kita dapat melihat bawah klien telah melakukan request ke arah server basis data. kemudian request yang di terima oleh server mongo klien akan diproses oleh mongoDB untuk selanjutnya akan di lakukan proses sharding atau memecah data request ke beberapa shard server pada pengujian ini akan di kirim ke first-shard dan second-shard sebagai server yang telah di kenali oleh mongoDB. Hasil dari sharding basis data akan tampak pada gambar 4.13 di bawah ini.

Gambar 4.7 Arsitektur Port Sharding

Pada gambar di atas kita bisa melihat bahwa dengan shard key bukuId yang merupakan key utama untuk memecah data apakah data akan masuk ke kluster frist-shard atau second-shard dengan metode hashed. Untuk membantu pembuktian data terkirim sesuai dengan hashed yang dimiliki ke masing – masing kluster peneliti akan memetakan data pada tabel 4.1 di bawah ini.

Tabel 4.1 Bagan Ilustrasi Pemetaan Sharding

mongodb	kluster	
	frist-shard	second-shard
"algoritma"	"algoritma"	"fisika"
"kalkulus"	"kalkulus"	"expedisi"
"agama"	"agama"	"elektronika_dasar"
"algoritma"	"algoritma"	"budaya"
"logika_dasar"	"logika_dasar"	"DuniaSophie"
"fisika"	"agama"	"DuniaSophie"
"expedisi"	"agama"	"DuniaSophie"
"elektronika_dasar"	"bahasa"	"algoritma_dasar"
"budaya"	"bahasa"	"algoritma_dasar"
"DuniaSophie"	"teknik_dasar"	"algoritma_dasar"
"algoritma_dasar"	"teknik_dasar"	"algoritma_dasar"
"algoritma_dasar"	"logika_dasar"	"algoritma_dasar"

Berdasarkan data pada tabel di atas dengan metode sharding data dengan nilai hashed yang sama akan selalu masuk di kluster yang sama sebagai contoh dengan shard key “algoritma” data akan selalu masuk pada kluster first-shard sedangkan dengan shard key “algoritma_dasar” data akan selalu masuk pada kluster second-shard. Berdasarkan pada uraian data tabel dapat diambil kesimpulan bahwa metode sharding ini dapat memecah data ke beberapa kluster yang telah terintegrasi. Maka dari itu menggunakan

5. Kesimpulan dan Saran

Berdasarkan pada hasil penelitian dan pengujian yang telah dipaparkan pada tahapan – tahapan sebelumnya dapat diambil poin – poin utama bahwa:

1. Sharding pada sistem basis data yang telah diteliti mampu menerapkan jenis skala horizontal pada sistem basis data berjalan yaitu menyebarkan data ke beberapa kluster server.
2. Penerapan sharding pada penelitian ini membuktikan bahwa dengan metode hashed

Daftar Pustaka:

- [1] D. P. Damayanti, “Analisis Performansi Sharding / Partitioning Pada Document-oriented Basis data (NoSQL),” 2012.
- [2] P. Maia, “Sharding by Hash Partitioning - A Basis data Scalability Pattern to Achieve Evenly Sharded Basis data Clusters,” Conference: 17th International Conference on Enterprise Information Systems, 2015.
- [3] A. M. Hidayatullah, “Perbandingan Performa Basis data Dengan Menggunakan Metode Replikasi Menggunakan Maxscale Pada Sistem Operasi Centos,” 2020.
- [4] T. Y. Tangkuman, “Analisis Pemanfaatan NoSQL Basis data Elasticsearch,” 2020.

metode sharding adalah solusi terhadap sebuah sistem yang berjalan ketika sudah memiliki data terlalu banyak sehingga memerlukan penambahan kapasitas terhadap media penyimpanannya. Hasil dari pengujian ini ialah metode sharding dengan menggunakan basis data NoSQL dengan jenis mongoDB merupakan solusi yang efisien dalam proses peningkatan sistem basis data horizontal. Besaran penyimpanan data akan selaras dengan jumlah kluster yang dimiliki oleh sebuah sistem.

mampu memetakan data sesuai dengan nilai hashed yang telah ditentukan.

3. Di dalam sistem sharding juga terintegrasi sistem replikasi yang mampu menangani ketika terjadi masalah pada sebuah sistem basis data

Hasil dari pemaparan poin – poin di atas di tarik kesimpulan dengan adanya sistem sharding dapat menyediakan perusahaan yang bergerak pada bidang Informasi dan teknologi untuk memfasilitasi sistemnya yang telah berjalan dan memerlukan peningkatan.

- [5] N. P. Siregar, K. R. W. S. dan A. A. G., “Analisis dan Implementasi Basis Data Terdistribusi Horizontal pada MongoDB untuk ClikKB BKKBN Regional Jawa Barat,” p. 11, 2018.
- [6] H. Krishnan, M. Elayidom dan T. Santhanakrishnan, “MongoDB – a comparison with NoSQL Databases,” International Journal of Scientific and Engineering Research 7(5), 2016.
- [7] Plugge, E., Membrey, P., & Hawkins, T., “The Definitive Guide to MongoDB. The NoSQL Database for Cloud and Desktop Computing”, 2010.